Bytemaniacos

Lo más clásico en informática y videojuegos



Nº 03

Concursos
CSSCGC 2008
Interface 2008
BASIC 2008

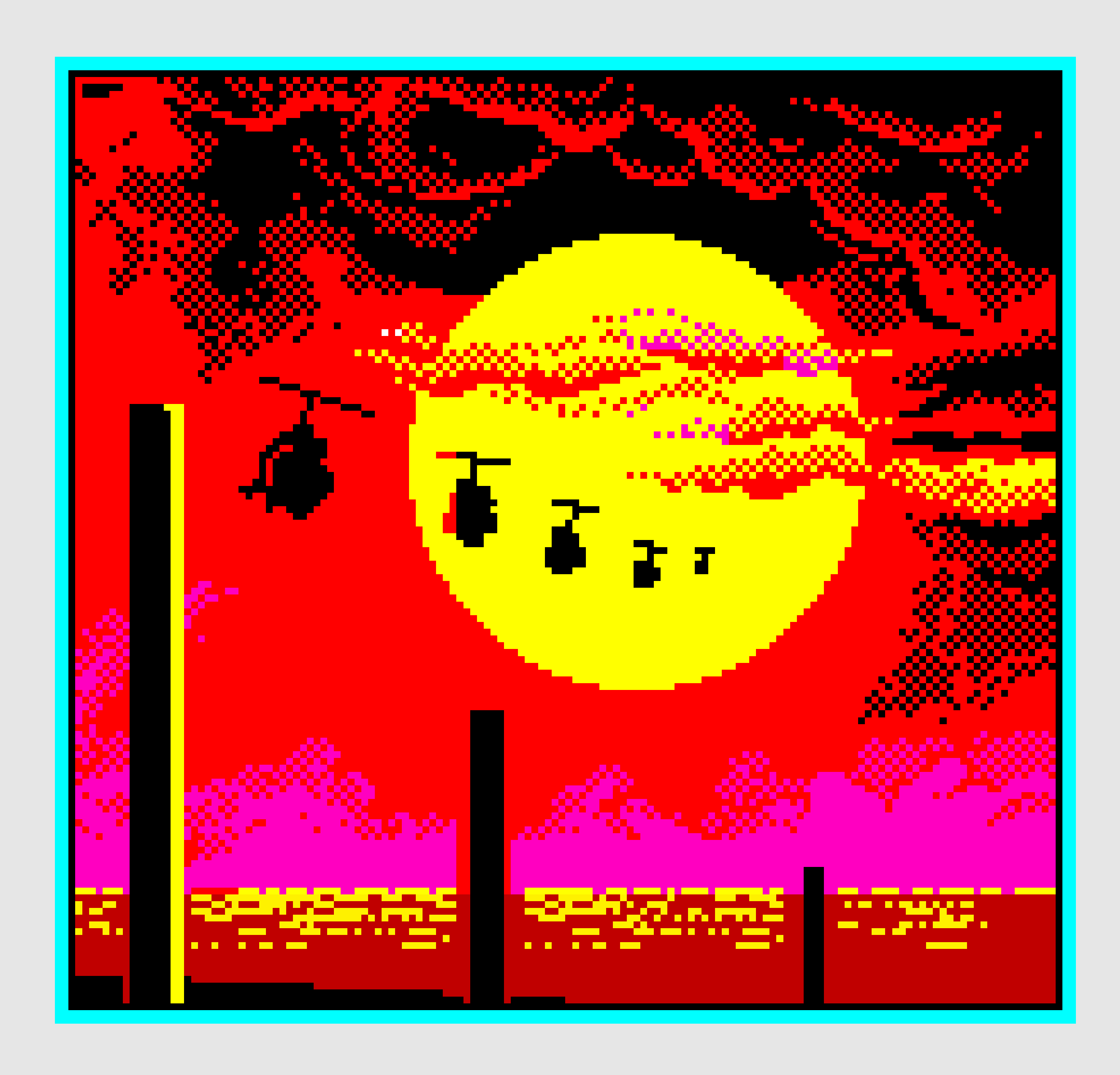
El archivo de Igor
RPG's en primera persona

Análisis
Farmer Jack
Isotopia
Small Games fo Smart M.
TFG Value Collection

Artículos
ZX Spectrum, entornos 3D
ZX Spectrum, Cartuchos

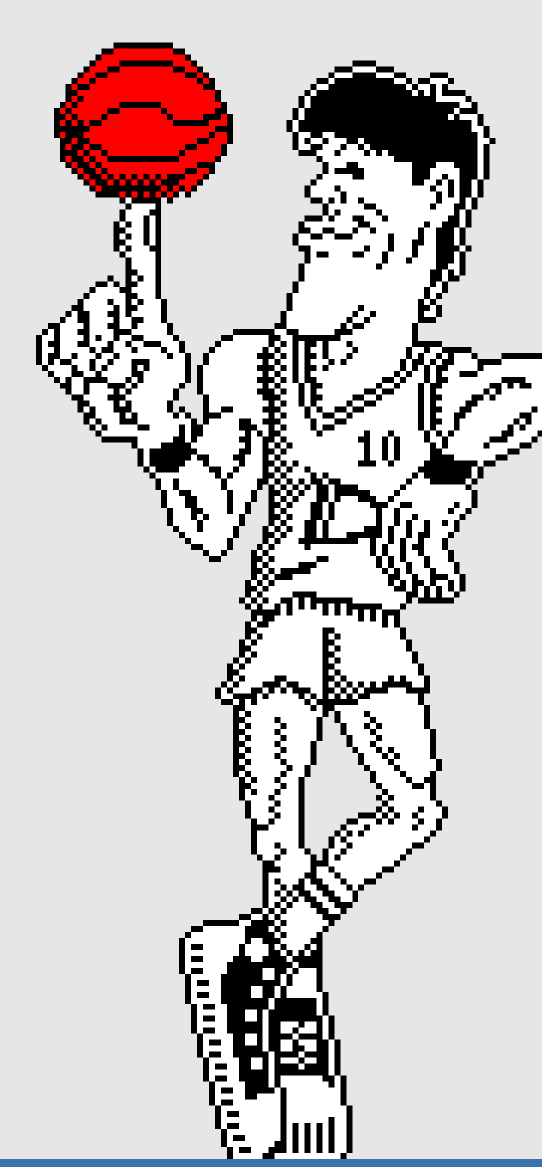
Retro
Kane

Índice



Editorial

"No existen más que dos reglas para escribir: tener algo que decir y decirlo."



Y van tres...

Este número es especial, y lo es por varias razones. En primer lugar porque tiene más contenido que los otros dos, principal motivo del retraso, en segundo lugar porque aparecerá poco antes de Retro Madrid 2008, y en tercer lugar porque comenzamos nuevos concursos.

Debería haber creado dos fancines, eso era lo fácil, pero haciendo caso de las sugerencias de algunos lectores he creado un volumen laaaaaaargo y completo. También he alargado un poco más el texto, para procurar contentar a aquellos lectores que aprecian la extensión de los últimos libros de George R.R. Martin

En esta ocasión no he puesto sección de hardware, ya que hay ración doble de programación (así, sin anestesia). He intentado que dichos artículos sean accesibles a todos los públicos: a los principiantes y con poco tiempo (BASIC), y a los que gustan de retos más elevados (ensamblador).

Pues nada, coged vuestro refresco favorito, unos snacks, y a pasar un buen rato leyendo las paridas y noticias que os trae un servidor.

Radastan

Viaje al Centro de la Tierra

Por muchos es conocido el proyecto de Topo Soft, que vió la luz para ordenadores de 8 y 16 bits. Un juego de manufactura exquisita, con varios niveles que mostraban en todo su esplendor la archiconocida obra de Julio Verne.

En ZX Spectrum gozamos de una buena versión, pero que desgraciadamente no llegó a tener todos los niveles de sus hermanos de 16 bits.

Pero el tiempo pone las cosas en su sitio, y es aquí donde aparecen dos personas con muchas ganas por corregir la injusticia.

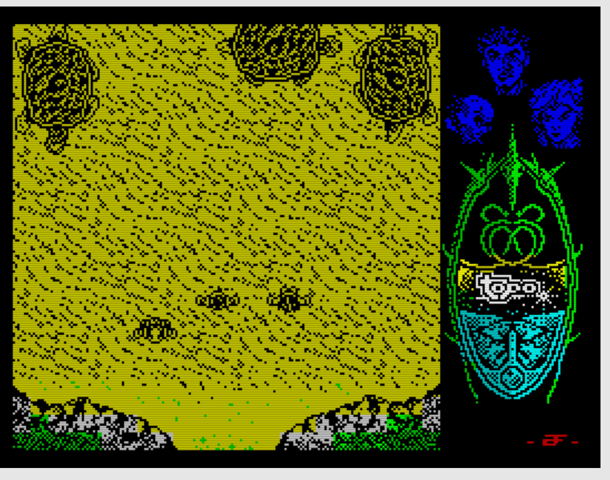
Han recibido numerosas críticas por ello, tanto por retomar el proyector como por la forma de hacerlo. Es posible que el resultado no guste a todos, o que la posibilidad de jugar a las nuevas fases por separado se considere un sacrilegio, pero no hay que restar el mérito por la labor realizada y el resultado.

La nueva fase 4 nos presenta una playa llena de tortugas, en la que nuestros tres protagonistas deben realizar malabares para superarla. Gráficamente no es la panacea, el desarrollo entretiene pero no llega a ser un nivel imprescindible, la verdad es que no es una gran pérdida para el juego original.

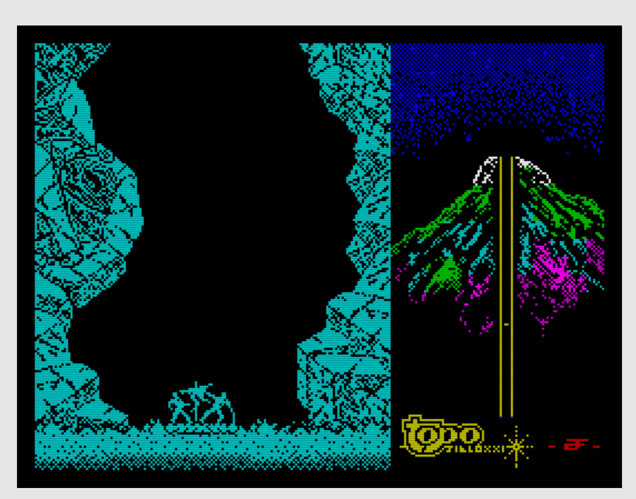
La fase 5 es todo lo contrario, aún siendo más sencilla (la verdad es que con un poco de atino se termina a la segunda), representa un pasaje crucial para el desenlace de la historia.

La posibilidad de jugarlos por separado nos da un sabor agridulce, ya que muchos esperábamos una versión extendida del juego original, pero al menos nos es posible probar como “bonus” las legendarias fases perdidas.

Felicidades a los autores.



El nivel de las tortugas no es el mejor del juego, pero divierte



El nivel final es imprescindible, el juego original debería tenerlo

Nueva línea de juego sencillos de CEZ

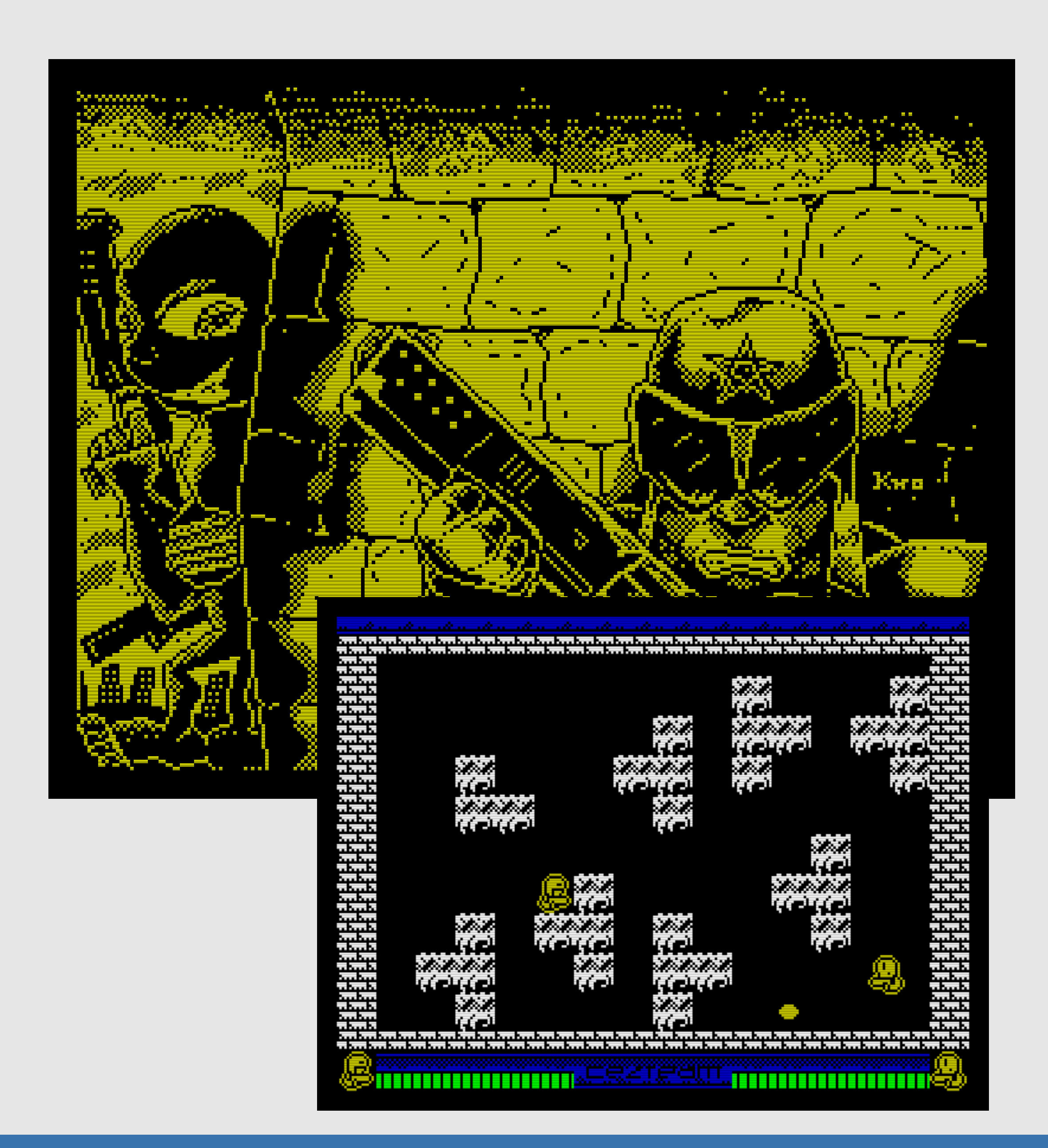
Computer EmuZone nos presenta su nueva línea de juegos CEZ minigames. Bajo este nombre encontramos juegos de rápido desarrollo, con el único objetivo de desarrollar ideas nuevas y sencillas.

La idea es realmente buena, ya que son juegos que de otra forma sólo tendrían cabida en mini concursos o competiciones de programación. Nos recuerda enormemente a iniciativas de antaño, como la de Dinamic, creando una línea alternativa de juegos denominada 100% Dinamic (que sólo llegó a sacar el juego Delfox)

El primer juego en aparecer es Judge Morry vs. Baldo Midget, un furioso juego de combate entre dos jugadores. Desarrolado por los Mojón Twins, nos plantea un campo de juego varible para que dos personas se maten a tiro limpio: sencillo y muy divertido. Destacar la pantalla de carga, perfecta, y el suave movimiento de los personajes. Un nivel técnico bastante bueno para ser un juego de este tipo.

Esperamos que las próximas obras sigan a este nivel.

http://computeremuzone.com







¿Quieres ver rarezas? No pierdas esta oportunidad

8 de Marzo de 2008, grabad dicha fecha en vuestra mente con letras de fuego. La mayor feria de retro informática de España celebra su nueva edición.

Si conoceis el tema no hace falta deciros más, no ir es perderse mucho, y si nunca habeis ido sería imperdonable perder la oportunidad.

Decenas de ordenadores de 8 y 16 bits, videoconsolas de todo tipo y época, chismes y más cosas.

Es imposible narrar en palabras todo lo que se puede ver y tocar allí, además de comprar.

Nosotros vamos a ir, con algunas ideas para disfrute de todos: mesas pegadas a la pared, cacharros funcionando y a la disposición del público, gente dispuesta a explicar lo que ves.

Este año nos vamos a divertir.

http://www.retromadrid.es






Grupo de Usuarios Amstrad
OCTOCOM
KAROSHI CORPORATION
Frikis
Frikis
Frikis
Frikis
ATARI
Frikis
Museo8Bit
Old Comput
Old Comput
Frikis
ZST Software
EMUFORGE
DREAMCAST-GP2X
DCiberia
ESCENARIO
RASTRILLO
Frikis
<- AULA DE CONFERENCIAS
Frikis
Mogollón de Frikis
Retro Madrid
ENTRADA
RetroAcción
Bytemaniacos
Bytemaniacos
Bytemaniacos

Onechip MSX

http://www.bazix.nl/onechipmsx.html - http://www.onechipmsx.com



BAZIX sigue adelante con los usuarios europeos

Con un año de retraso BAZIX ha dado nuevas noticias sobre el futuro del Onechip MSX en Europa.

Tras la puñalada trapera de D4 Enterprise (nadie esperaba lo contrario) se está buscando un nuevo fabricante. El objetivo es lanzar una versión mejorada respecto a la japonesa sin un aumento significativo en los costes.

Esto significa más espera, pero un premio final mucho más jugoso.

De igual forma anuncian que los que teníamos reserva tendremos preferencia, y que se admitirán más reservas para los que quieren uno.

Recordamos que este invento es un MSX2 con 1 MB de memoria, 128kB de VRAM, Kanji-ROM, MSX-Music, MSX-DOS2 con FAT 16, dos puertos USB, MEGA SCSI con tarjetas SD, salida VGA, doble cartucho... un sueño perfecto.











Knight Lore aterriza en Atari XL/XE

Krzysztof Dudek, Mario Krix, y Michal Radecki son tres personajes que en muy poco tiempo están moviendo la escena de Atari 8 bits. Este grupito ya dió la nota anteriormente con Jet Boy, conversión del conocido Lunar Jet Man de ZX Spectrum, y ahora se han atrevido con nada menos que Knight Lore, conversión realizada desde BBC Micro.

Si un servidor no se equivoca se han sacado las melodias de la chistera, y son realmente buenas al igual que los FX, proporcionando una buena sensación de conjunto en todo el juego. A los usuarios de ZX Spectrum les será raro escuchar música en este juego, pero queda realmente bien.

Eso si, una pena que la pantalla de presentación no use más color, podrían haber conversionado la de ZX Spectrum, pero tampoco queda mal.

Una grata sorpresa para Atari 8 bits, ¿Se atreverán con Batman?

http://atari.fandal.cz/detail.php?files_id=5710

Con un buen par

Hay gente valiente, hay gente que se lo curra, pero sólo unos pocos tiene ese “par” necesario para dar el campanazo.

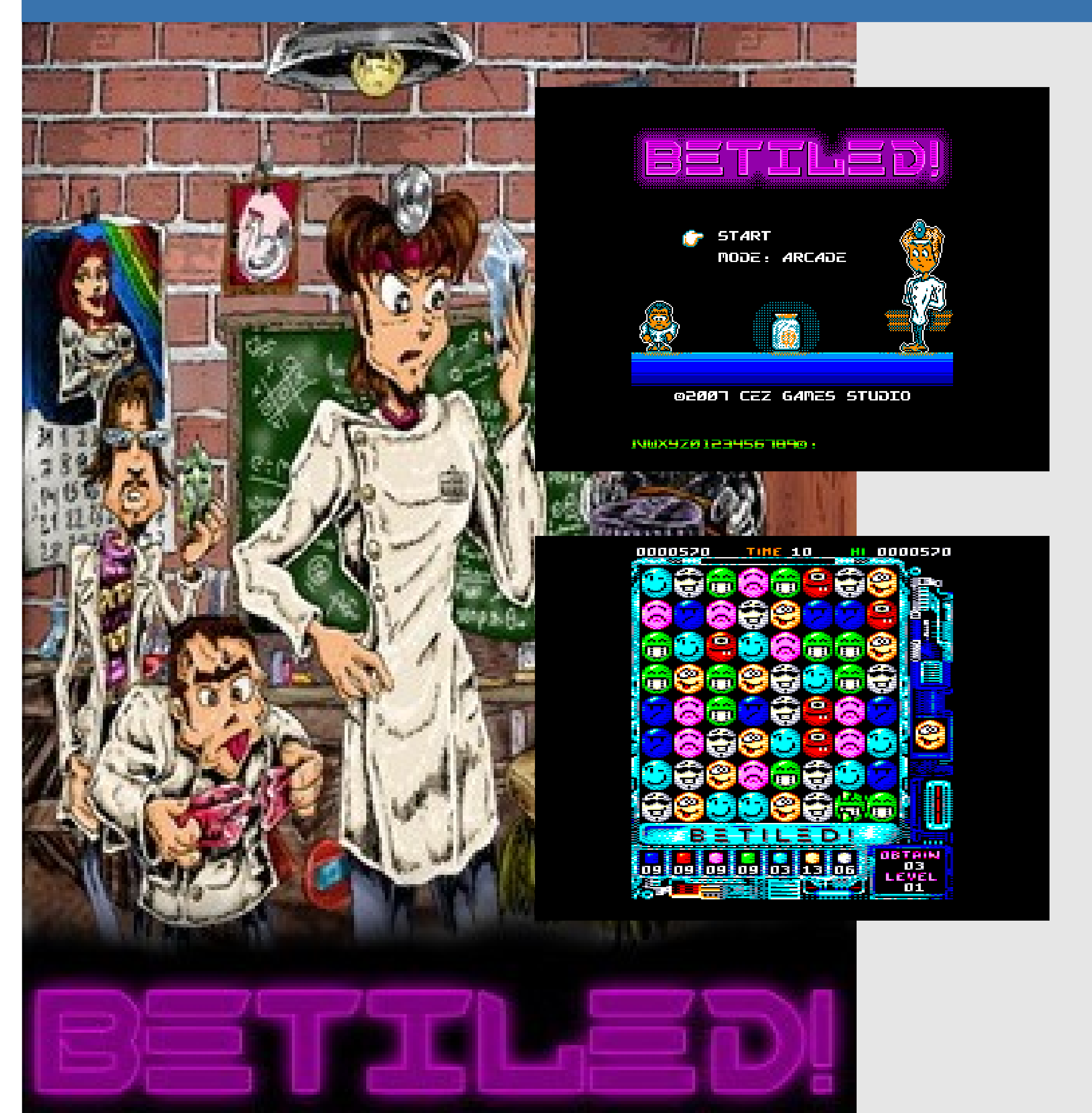
Lo que veis a la derecha no son imágenes de Elvira para Amiga o C64... son de un humilde Amstrad CPC, y su autor Markus (DevilMarkus para los amigos).

El juego está en un avanzado estado de desarrollo, con muchas posibilidades de estar finalizado antes del verano, y está recibiendo ayuda de otras personas para su consecución.

Afortunadamente se ha olvidado el modo 1 y se ha escogido finalmente el modo 0 para los gráficos, lo que significa unos gráficos mucho más coloridos que en la beta anterior. Otra novedad son las animaciones, hay escenas de vídeo como en 16 bits, y todo se mueve realmente bien. Si quereis babear un poco:

http://cpc.devilmarkus.de/elvira/elvira.html





Betiled! casi listo para Amstrad CPC

Es muy previsible que en Retro Madrid podamos disfrutar de la versión CPC de Betiled! En esta ocasión el apartado técnico será espectacular, aprovechando al 100% las posibilidades de la máquina.

Comenzando por un menú sorprendete, que usa 14 colores en mode 1 (recordemos que la teoría dice que no se pueden emplear más de 4 colores), animaciones, efectos de luz, scroll de texto pixel a pixel multicolor, etc. Y empleando además overscan para superar los límites de la pantalla.

Betiled! Ya fué lanzado para ZX Spectrum y MSX, siendo ambas versiones muy elogiadas en la escena nacional e internacional por su adicción, jugabilidad, y por intentar aprovechar cada sistema y no ser una mera conversión.

Nos quitamos el sombrero ante tal despliegue de medios, y esperamos que las próximas producciones tengan tanto mimo y dedicación.

http://www.computeremuzone.com

CSSCGC 2008

http://reptonix.awardspace.co.uk/sinclair/csscgc2008/

El peor juego del mundo tiene premio

Crear el juego peor realizado, más absurdo, sin sentido, injugable... ese es el objetivo.

Año tras año las news de comp.sys.sinclair organizan la Crap Game Compo, y en cada edición vemos auténticas barbaridades.

Pero este año hay gran competencia, en parte debido al pique de la escena española. Juegos como Eat Shit!, Crysis, o el simulador de primera comunión de los Mojón Twins, ya son clásicos de este año por derecho propio. Está por ver como se tomarán los ingleses el humor español, pero nuestro pais está poniendo el listón bien alto.

Otra novedad es que los españoles estamos empleando gráficos llenos de color, y es que el BASIC con sus bloques pueden dar mucho de si, aparte de dar una apariencia más bizarra todavia a los juegos del concurso.

Destacamos la obra magna de los Mojón Twins, su simulador de primera comunión, con gráficos enormes, pantallas dignas de Van Gogh, y un uso exquisito de los decorados.

A muchos ha sorprendido también la conversión de Crysis, ya que presenta un FPS perfectamente jugable desde lenguaje BASIC.

Perded unos minutos, merece la pena.



Eat Shit! Nunca comer mierda fué tan divertido



Podrás recordar hasta tu primera comunión

Radastan os presenta su apuesta segura

La CSSCGC no tiene porqué recibir exclusivamente juegos injugables, pueden ser juegos que simplemente por su temática sean completamente ilógicos. Es aquí cuando una mente piensa en una idea realmente ilarante, que no tiene cabida de ninguna otra forma que no sea por el placer de ver una chorrada en pantalla... así nace “Braveheart's Torture: a dance game experience”. Sólo con el título uno ya sabe que es radicalmente absurdo, porque mezcla tortura y baile.

Y así es, el primer juego de tortura musical de la historia, y uno de los primeros juegos musicales para ZX Spectrum (que uno sepa).

En unos días vereis la tortura más divertida de la historia, con unas posturas del ejecutor antológicas.

¿No os parece glorioso?

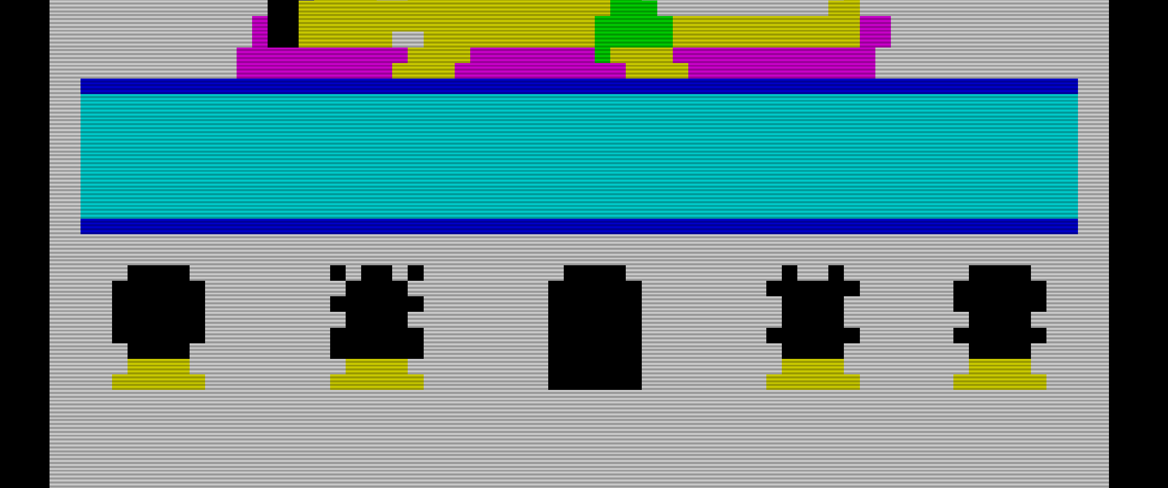
“El primer juego de tortura musical de la historia”.



KEYS

C - Cut
D - Dulce Jubilo
E - Extract organ

Press Space to start execution



Interface 2008



Rescatamos los cartuchos para ZX Spectrum

En los inicios del ZX Spectrum se pensó en usar cartuchos para mejorar la experiencia de los juegos. Es un formato cómodo y de acceso instantáneo, evitando el tedio de la carga desde cinta. Pero fué el precio, y la necesidad de tener una interfaz compatible (Interface 2) lo que llevó el invento a la ruina.

Pero los años han pasado, realizar una interface 2 y los cartuchos es muy económico, y el soporte es ideal debido a que las cintas van a su desaparición. De aquí la idea de este concurso, que premiará al mejor juego presentado.

Pueden participan aquellos videojuegos que cumplan estos requisitios:
- Ser de nuevo desarrollo
- Estar programados para ser usados como un cartucho ROM de 16 Kb de tamaño máximo
- Usar los recursos de la máquina original o del Interface 2 (Joystick), sin hardware adicional
- Estar realizados para funcionar en un ZX Spectrum con 16Kb de RAM

El único formato admitido es el de cartucho ROM (un fichero binario .ROM con 16 Kb de tamaño). Los participantes deben enviar su juego, con un fichero de instrucciones como mínimo, a radastan@gmail.com. Se ruega mandar otro e-mail sin ficheros adjuntos indicando que se ha enviado una participación, para verificar la recepción.

El plazo de enrega finaliza el 23 de Diciembre de 2008, inclusive, y el ganador se llevará un cartucho conmemorativo.

CONCURSO DE PROGRAMACION DE VIDEOJUEGOS

BASIC 2008



Por sexto año consecutivo, que se dice pronto, llega el concurso de videojuegos en BASIC para ZX Spectrum. Pero este año va a ser muy especial, será un “uno contra todos”. Radastán en persona participará en todas las categorías y los participantes podrán elegir el método de votación, para asegurar la imparcialidad. Así que teneis dos retos: ganar el premio... y derrotar al mecenas del concurso.

Sobre el concurso

En este concurso participarán aquellos videojuegos que cumplan los siguientes requisitos:

- Ser de nuevo desarrollo
- Estar programados integramente en lenguaje BASIC, sin rutinas en ensamblador
- Usar exclusivamente los recursos de la máquina original
- Estar realizados para ZX Spectrum en cualquiera de sus variantes (16K, +3, etc).

Plazo de entrega

El concurso finalizará el día 30 de Diciembre, inclusive, para que los últimos días de Navidad sean aprovechados por todos en probar los juegos presentados.

Los trabajos deben mandarse en formato .TZX .TAP o .DSK a kurdoman@gmail.com

Una vez presentados todos los participantes se les consultará la forma de votar.

CONCURSO DE PROGRAMACION DE VIDEOJUEGOS

BASIC 2008

Categorias

Programación BASIC pura: se prohibe el uso de llamadas a rutinas de la ROM directamente, el uso de peek o poke, y no se permiten GDU's. Sólo se admite un único bloque BASIC.

Programación BASIC libre: se puede emplear cualquier recurso sin usar lenguaje ensamblador. Se admite usar tantos ficheros como se desee.

Programación BASIC 7, igual que BASIC puro y con la limitación adicional de emplear hasta 7 líneas de código como máximo. En esta modalidad sólo se admite el uso de poke para usar gráficos definidos por el usuario (GDU's), y también se limita el uso del símbolo ":" a 10 por línea como máximo.

En la segunda categoria se pueden usar las rutinas de la ROM como se desee, redefinir el set gráfico o el de GDU's, y hacer todas las triquiñuelas que uno quiera. El objetivo es realizar el mejor juego posible dentro de las posibilidades del BASIC.

Premios

BASIC puro y BASIC 10: lote de videoconsolas y juegos.

BASIC Libre: ZX Spectrum +3





Rpgs en primera persona

¡Ya es de día en el castillo!

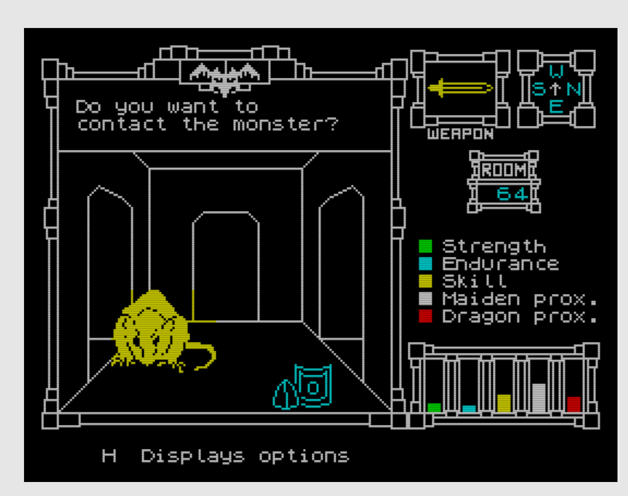
¿Han descansado? Espero que el fantasma del Conde Margoz no les haya interrumpido el sueño, su decapitación aún le mantiene furioso. Les puedo preparar unas deliciosas tostadas con queso de Buitre, una rareza realmente exquisita. ¿No? ¿prefieren fruta nuevamente?

Pasen al comedor, junto al fuego, les narraré una historia mientras les sirvo un vaso de zumo de hiedra venenosa.

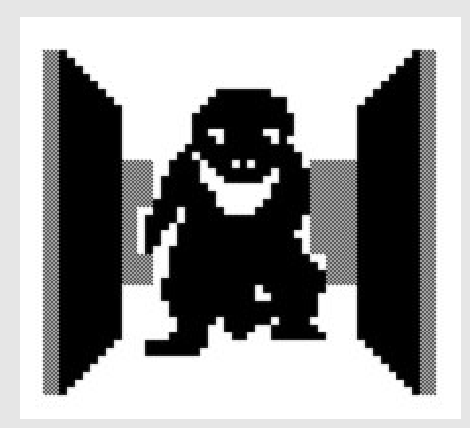
Las mazmorras lucen mejor en 3D

Con la llegada de los 16 bits, y el archiconocido Dungeon Master, el género del rol conoció una época dorada de pasadizos y enigmas en artísticas 3D. Pero los 8 bits ya conocían el género, y las mazmorras no estaban nada mal...





Los enemigos estaban bien representados en Dragonsbane



3D Monster Maze propuso un laberinto con un dinosaurio

3D Monster Maze

No podemos hablar de este género sin remontarnos a sus orígenes, y este juego es la semilla que abrió el camino.

No es un RPG propiamente dicho, no hay equipo, armas, comida, o variedad de enemigos, pero expuso un laberinto en 3D (las mazmorras no dejan de ser eso) y un enemigo que deseaba matarnos (idem de idem para los RPG).

Apareció para el ZX81 en 1981, valga la redundancia, y supuso un éxito inmediato por su originalidad, los gráficos, y el miedo que daba el dinosaurio avanzando sobre nosotros por aquel entonces (ahora da risa).

Entonces fué cuando el resto de programadores ató cabos. Así fué como los laberintos / mazmorras en 3D y en primera persona nacieron.

Evidentemente los primeros juegos no mostraban combates en tiempo real, y mucho menos en primera persona.

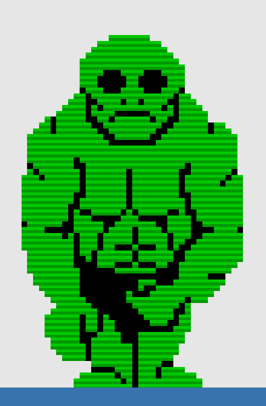
Dragon's Bane

Ya tenemos la semilla, un laberinto en 3D, y la forma de movernos por él. Es entonces cuando M. Preston y su equipo piensan en ampliar la idea: en vez de un laberinto ponemos una mazmorra, en vez de un enemigo persiguiéndonos ponemos varios que aparecen por zonas, añadimos objetos, comida... y ya tenemos un RPG en 3D y en primera persona para ZX Spectrum.

Este juego no es la panacea del género, pero es muy adecuado para los principiantes por un motivo: es muy sencillo. El quipo es muy sencillo de usar, la comida te la va pidiendo el ordenador, y los combates los lleva también el ordenador.

Gráficamente es algo simple pero agradable, se deja jugar muy bien, y un simple 48K sirve para disfrutarlo.

Recomendable.



Ellak's Tomb

Un año más tarde los usuarios de C64 recibieron un RPG muy peculiar. En este caso la visión de las catacumbas era igual de simple, los combates pasan a ser en grupo, y podemos elegir el oficio de los protagonistas.

Otra novedad es que los combates se vuelven estratégicos, ya no es uno contra uno, y cada miembro del grupo puede realizar una acción. Esto, añadido al hecho de poder tener oficios, añade un valor incalculable al juego. Los combates se viven con más intensidad, puedes lanzar magia, curar a los heridos, atacar mientras proteges al sanador, etc.

El juego llegó a tener bastante tirón, incluso se tradujo al español, y los amantes de los juegos de ROL de mesa comenzaron a descubrir que los RPG de ordenador empezaban a parecerse a los de verdad.

Pero sólo fué el principio.

Bard's Tale

En 1985 los cimientos de los RPG se tambalearon, surgió la leyenda.

Bard's Tale es el primero de los RPG's modernos, contiene todo lo que un aventurero puede desear: razas, oficios, características, niveles, y un largo etcétera.

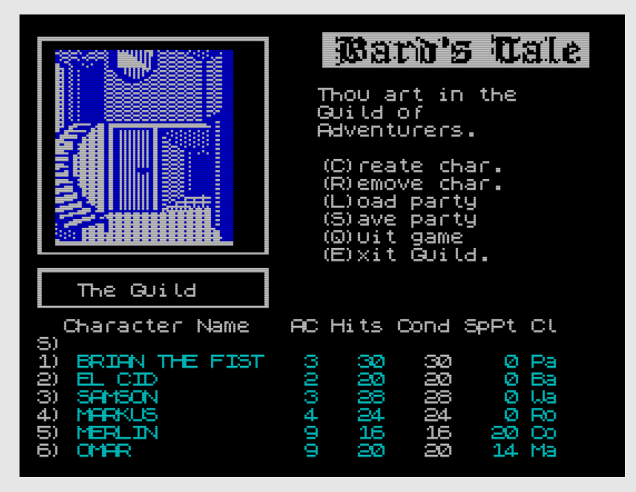
Se basó en las reglas de Dungeon's & Dragons y en muchas de las cosas que hicieron de Wizardry en 1981 un juego de culto (Wizardry es un RPG en tercera persona).

La visión en primera persona es mucho más compleja, al igual que el juego en si. La profundidad del juego es tal que en Commodore 64 usaba varios disquettes y en ZX Spectrum varias cintas (dos discos en la versión +3).

Los usuarios de Spectrum tuvieron que esperar tres años para su versión, mientras C64 recibía la tercera parte.



En Ellak's Tomb los combates son en grupo



Bard's Tale es realmente complejo, no es para novatos





El juego tiene una buena secuencia de presentación

En 1987 los usuarios de Atari 8 bits y los de C64 recibieron un juego grande en todos los sentidos: tenía una intro currada, los gráficos eran bestiales para el género y la época, largo... y terrorífico para las disqueteras.

Su novedad estriba en que no se lanzó un juego, sino varios, cada uno ambientado en un mundo distinto (la ciudad, la mazmorra, etc). La historia gira entorno a una abdución extraterrestre, en el cual se te manda a una época distinta según el juego que compres (originalmente hiban a ser 6 épocas, se quedaron en dos). El juego casi más que un RPG es casi una videoaventura, pero poseemos las características del personaje, el sistema de exploración las tiendas... es un conjunto raro, pero realmente divertido.

El que nos ocupa nos deja en unas mazmorras, y ya no se mueve todo a trompicones. Si uno avanza la mazmorra avanza poco a poco, no así al girar, lo cual al menos nos deja una sensación más realista en 3D y hace que nos orientemos mejor en el mapa.

Un único apunte: armaros de paciencia, cambiareis de disco cada 2x3.



Las mazmorras se mueven con suavidad si avanzas



En 1986 vimos nacer los juegos basados en la serie Might and Magic, que actualmente siguen con buena salud.

El “Libro primero” fué disfrutado por usuarios de MSX y C64, y casi tiene todo lo que un RPG en primera persona debe de tener (salvo los combates).

Al año se lanzó al mercado de los 16 bits “Dungeon Master” el que realmente es un RPG en primera persona como tal, ya que los combates son en tiempo real.

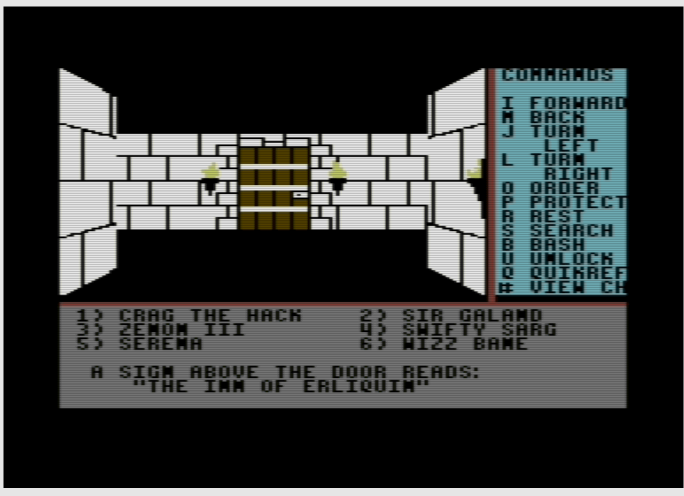
No obstante, sin ser puristas, hay que admitir que el juego que nos ocupa es un juegazo en toda regla.

Llevamos a un grupo de 6 personajes en busca de una especie de santo grial, y a lo largo del juego la historia se desarrolla con su trama, para acabar desenmascarando al malo maloso y finalizar pasando por el portal a otros mundos (es decir, que juegues la segunda parte).

Los gráficos son muy vistosos, bien diseñados, con variedad, y además no se limitan a mazmorras (sales a bosques, valles, etc).

En 1989 se lanzó la segunda parte, mejorando aún más los gráficos pero manteniendo el sistema de combate. Dicha parte incluyó uno novedoso sistema de ”vejez”, en el cual los personajes morían al alcanzar la edad de 75 años aproximadamente. También se añadieron los combates masivos de Bard's Tale, y podías encontrarte en un poblado Goblin contra 300 enemigos.







Llegó la bestia

Año 1990, mientras España miraba por TV los mundiales de fútbol en Italia, los roleros de 8 bits quemábamos las pestañas ante otro evento. El gran Bloodwych visitaba nuestros ordenadores (incluso el Amstrad CPC), conservando todo lo que hizo grande las versiones de 16 bits.

El primer RPG en primera persona en permitir dos jugadores simultáneos (no recuerdo otro anterior), con los mapeados más inmensos que podais ver en el género, y el primero de combate en tiempo real para 8 bits. El sistema de control con el teclado es muy bueno, teniendo en cuenta que el original es con ratón. Este hecho le hace muy sencillo de manejar, sin tropecientas teclas de acción y movimiento, incluso permite usar un simple joystick.

La historia nos pone como los típicos campeones, osea, te ha tocado salvar el reino por los cojones del rey. Nuestro objetivo es derrotar al malvado Zendick y al Señor de la Entropia (mooooola). La historia te hará buscar las gemas dispersadas por varias torres para abrir la zona final, y os puedo asegurar que en modo de un jugador es absorvente, pero a dos jugadores es bestial.

Jugando a dobles podemos pillar por sorpresa a los enemigos, atacarles por las espaldas, etc. Si encima se añade que habrá grupos enemigos como los nuestros, imaginad la diversión.

Y que mejor manera de despedir este día en el castillo de Frankestein que con la musa del doctor... Elvira no fué exclusivo de máquinas mayores, gracias a la popularidad del C64 se pudo disfrutar de una excelente versión de 8 bits que no andaba muy a la zaga de su versión mayor.

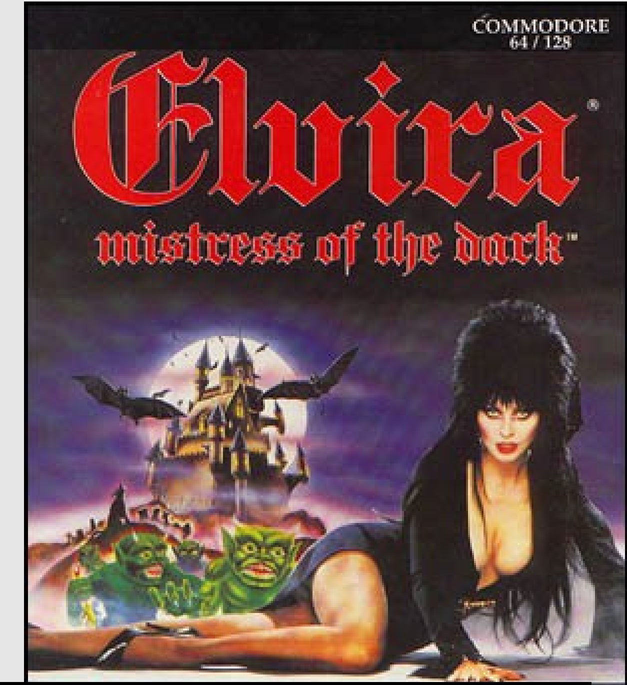
Con grandes dosis de sangre, enigmas, y un buen par de motivos para ver a Elvira, este juego se ganó por derecho propio ser un clásico en la historia de las videoaventuras. Y es que no se trata de un RPG, pero lo mencionamos aquí debido a que, al igual que hay que ver los orígenes, hay que ver cual es la evolución posterior.

Tiene el sistema de exploración de los RPG en primera persona, incluso sus combates, pero no es un RPG. No hay profesiones, ni otras muchas cosas clásicas del género.

Tras Elvira, los usuarios de C64 pudieron disfrutar de otros tres RPG bastante decentes: Shadow of the Evil (1993), Boom (1996), y Newcomer (2001). Este último englobado también en el género de la videoaventura más que en el RPG, y con un nivel visual muy alto.

No obstante Elvira y su segunda parte eclipsaron todo lo demás en calidad, sobre todo su segunda parte, y nada fué igual desde entonces. Se puede decir que es lo mejor que se puede jugar en 8 bits dentro del terreno de la aventura (con permiso del genial Bloodwych), tanto por historia como por desarrollo.

Y esto es todo, ahora dejo a los señores que puedan pasear por nuestra magnífica sala de tortura, donde podrán ejercitarse y realizar algunos estiramientos realmente tonificantes.







Una agradable sorpresa

Farmer Jack ha aparecido de pronto, sin llamar la atención, y nos ha dejado a muchos con un excelente sabor de boca. Se trata de un juego sencillo, de los que es para pasar el tiempo, pero cuando juegas ya no puedes parar.

Está basado en el juego Pac-Man, pero con "esas cosillas" que lo hacen único.

El juego

Imaginad por un momento el clásico Pac-Man, pero cambiemos a nuestro orondo protagonista por un tractor. Añadamos algunas paredes móviles, que podemos mover nosotros pero no los enemigos, y algunos detalles más.

Con esta simpleza el juego cobra una nueva dimensión, ya que se vuelve muy estratégico: puedes encerrar a los enemigos, cortarles el paso, etc.

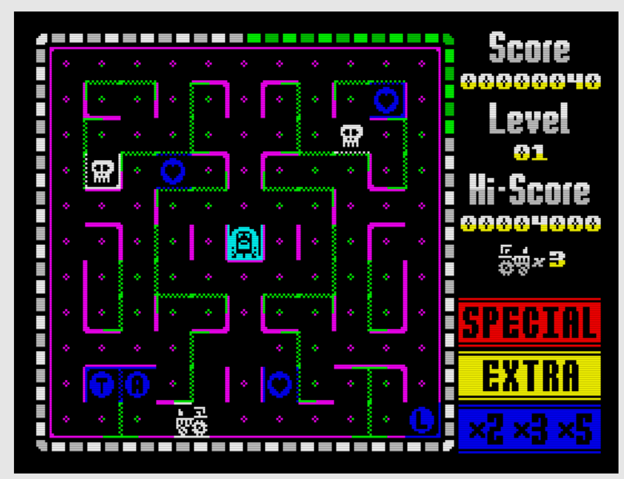
Es realmente divertido ver como te persigue un enemigo y en un plis-plas le cortas el acceso al tractor, o intentar una encerrona para apartarles a una zona ya recogida. Todo un acierto.

Gráficos

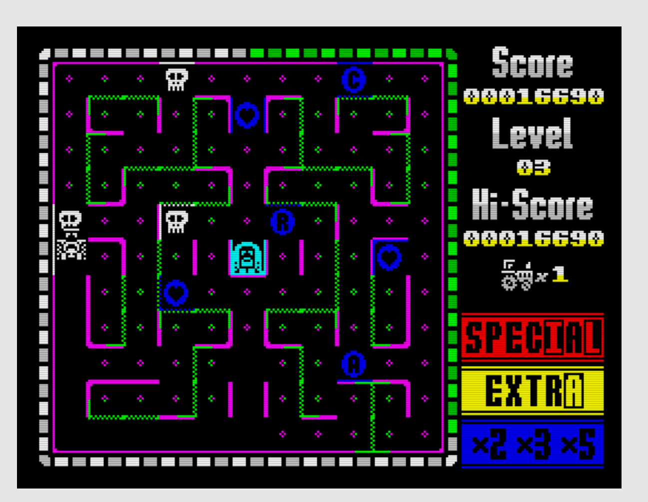
Empezamos por una pantalla de carga llena de color y viveza, una buena muestra de cómo hacer algo vistoso. El menú muestra un gran logotipo a color, y el resto de detalles son de agradecer , como las espigas de las puntuaciones.



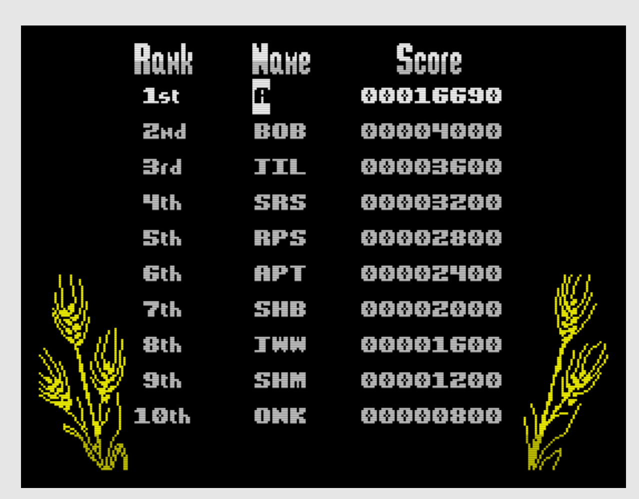
La pantalla de presentación es agradable y llena de color



Las paredes móviles jugarán un papel crucial



Las calaveras nos matarán si las tocamos



Hasta la pantalla de puntuaciones se ha cuidado

El juego en si tiene unos gráficos correctos, ya que deben ser pequeños por necesidad, pero se mueven con mucha suavidad. No osbtante podrían haberse mejorado en ciertos aspectos.

Melodías para tus oidos

El terreno musical está solventado con mucho acierto. La música envuelve cada aspecto del juego y la melodía es muy agradable. La verdad es que el juego no sería lo mismo sin su música, y además es muy pegadiza, de las que se te queda en la cabeza aún después de jugar.

Para mojar pan

Podemos finalizar diciendo que Farmer Jack es un juego de los que no pasan los años, seguirá siendo divertido por siempre. Si te gustan los arcades y pasar un rato divertido sin muchas complicaciones, este es tu juego.

Gráficos	7
Audio	9
Juego	9

Nota 8

Puedes descargarlo gratuitamente en:

http://www.worldofspectrum.org/infoseekid.cgi?id=0018873

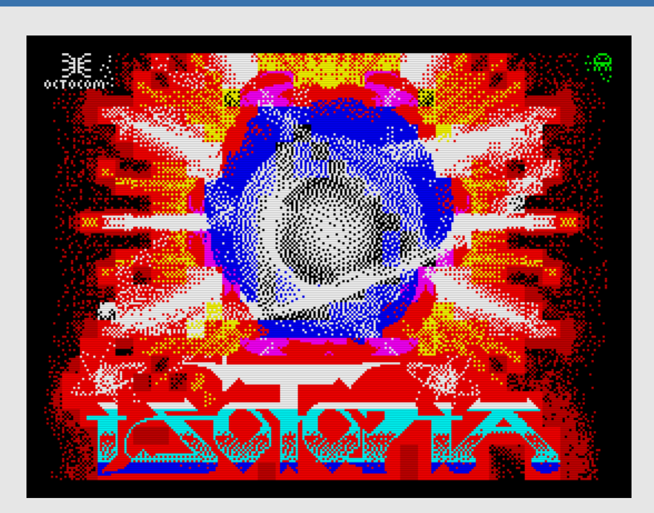




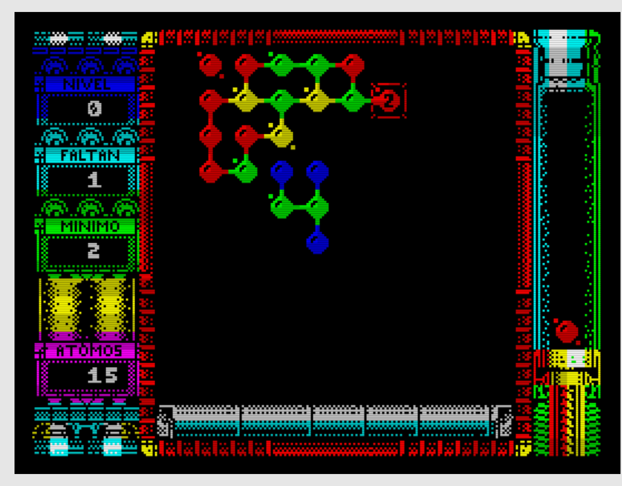
Octocom ha demostrado que con ideas sencillas y mucha paciencia se pueden hacer juegos realmente buenos. En esta ocasión han querido rendir homenaje a un clásico de Psygnosis, y de paso realizar su primer puzzle.

Nunca hacer moléculas fué tan divertido

Soy reacio a los puzzles, lo admito, pero me he esforzado a tope por sacarle el jugo a esta producción y así realizar un análisis lo más objetivo posible. En las primeras partidas no comprendí la forma de jugar, pero cuando lo entendí me vi sorprendido por su sencillez y lo divertido que era todo. Y es que Isotopia es un juego en el que no existe el amor a primera vista, hay que poner de tu parte para comprenderlo en toda su extensión y llegar a captar todo lo que ofrece.



La pantalla de carga es muy mejorable



No podemos dejar átomos sueltos para pasar de nivel

Gráficamente correcto

Los gráficos no son el punto fuerte del juego, y la verdad es que son muy mejorables en varios aspectos. La pantalla de carga pedía a gritos la misma que la carátula de la cinta, y a algunos detalles, como el menú o los marcadores, les falta ese toque final.

El juego en si está muy bien realizado, ya que los átomos son sencillos y muestran con claridad los "radicales libres" que poseen. Y la verdad, eso es lo más importante en un juego como este.

La música, a la altura

Pero hablamos de un puzle, donde prima la concentración, y la música es un apartado esencial. Beyker ha realizado un gran trabajo (va camino de ser nuestro Mozart de 8 bits), y su trabajo se hace notar. Sólo ha faltado una opción para controlar el volumen de la melodía, y algún FX para aderezar la acción.

Gráficos	6
Audio	8
Juego	8

Nota 7

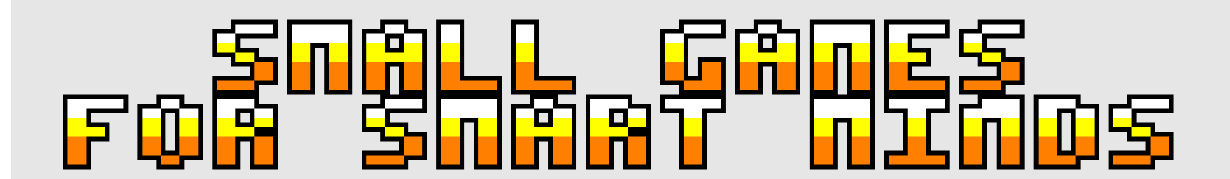
Resumiendo, un juego que no será del agrado de todos, pero al que no hay que quitarle su mérito.

Es divertido, sencillo de jugar cuando le cojes el truco, y puedes descargar incluso un editor para hacer tus propios niveles (todo un detallazo).

El problema es que el ZX Spectrum está saturado de puzzles y juegos sencillos últimamente, pero eso debe hacernos ver la calidad de los mismos.

Ahora a esperar ese juego de acción que nos han prometido...

http://www.octocom.es/juego06.html



Para los amantes de darle al coco

Small Games for Smart Minds es una recopilación de pequeños juegos para Amstrad CPC, todos ellos de naturaleza simple y enfocada a darle vueltas a la cabeza. La idea no es mala, con una sóla carga tienes 3 Juegos, pero el resultado no es todo lo bueno que debería ser.

Gráficamente es sencillo, siendo Sokoban el más vistoso de los juegos, pero falla en algo crucial: todo se mueve a bloques. En las Damas o el Othello puede no se importante, pero en Sokoban es realmente molesto al principio.

Pero lo más sangrante es el tema del sonido: nada de nada, y eso es imperdonable. Comprendemos que no haya música, pero unos “blips” al menos le hubieran dado algo más de salsilla, incluso eso es posible realizarlo con el BASIC sin mucho esfuerzo.

Un juego con carencias técnicas, que al menos cumple con su objetivo: es divertido.

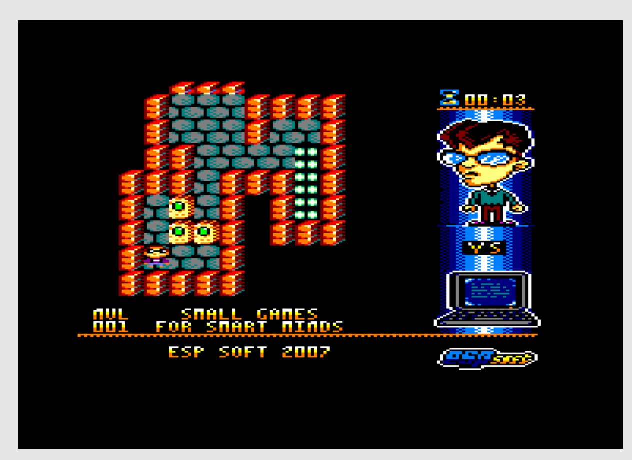
Gráficos	7
Audio	0
Juego	7

Nota 5





La pantalla de carga es genial, aprovecha el modo 0 a tope



Sokoban es el juego más entretenido con diferencia



El juego de ski es uno de los más divertidos



Jihad nos pone en la piel de un piloto de caza

The Fall Guy Value Collection

Otro recopilatorio, en esta ocasión para Commodore 64 y con 5 juegos muy dispares entre si. Nos recuerda mucho a los típicos juegos de Atari 2600, pero con mejores gráficos.

De entre todos nos quedamos con el de Ski y el de aviones, el resto son tirando a regular (el de star trek es un penoso matamarcianos).

El sonido y las melodias son bastante aceptables, en algunos casos hasta dan más sensación de juego añejo. Todo recuerda a los primeros juegos de Commodore 64.

Y poco más podemos decir, es la típica recopilación para entreteneerse un rato y poco más, difícilmente nos llegaremos a enganchar más de una tarde, pero al menos pasaremos un momento divertido.

http://www.redesign.sk/tnd64/friends/TFG_Value_Collection.zip



Star Trek es un insulto a los fans de la serie

Gráficos	6
Audio	6
Juego	6

Nota 6

Entornos 3D en lenguaje BASIC

Motor gráfico FPS/RPG en primera persona para ZX Spectrum

El BASIC de los ordenadores de 8 bits siempre se ha asociado con lentitud y limitaciones en lo que respecta a videojuegos. No obstante, si se piensa un poco en dichas limitaciones y se programa con ellas en mente es posible crear pequeñas obras de arte.

En la presente edición de la CSSCGC se ha presentado un pequeño juego en BASIC que simula ser un port de Crysis para ZX Spectrum. Como es lógico se trata de una broma, que sin embargo funciona y es perfectamente jugable.

Así que, a petición popular, vamos a comentar como se puede realizar un entorno 3D creible en el BASIC del ZX Spectrum de una forma sencilla y sin necesidad de usar lenguaje máquina.

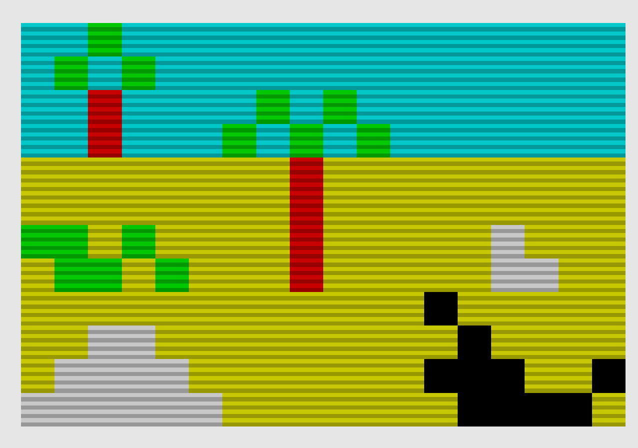
Pasen y vean.

Los gráficos

El ZX Spectrum posee una pantalla dividida en pixels, 256x192 para ser exactos. Dicha pantalla se divide en 32x24 celdas de 8x8 pixels, llamadas caracteres en BASIC y accesibles mediante el comando PRINT.

Lo primero que debemos hacer es plantearnos un límite realista en cuanto al tamaño de nuestro visor de pantalla, porque si pretendemos realizar un juego en BASIC y en 3D no podemos hacerlo a pantalla completa ni por asomo.

De igual forma debemos limitar el número de objetos en pantalla, ya que si ponemos mucho escenario todo se va a dibujar a paso de caracol, además de crear confusión.



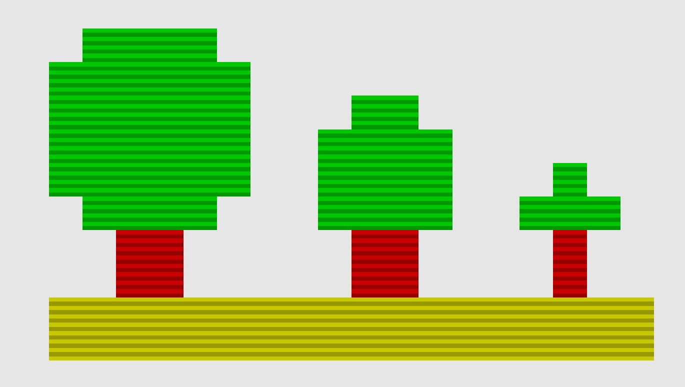
Crysis en ZX Spectrum, una coña en BASIC que funciona

Por todo esto, la solución más rápida y fiable es usar un visor de 9x6 caracteres y 9 objetos en pantalla (de decorado). No es gran cosa pero es un comienzo.

La anchura impar obedece a la necesidad de dividir la imagen en horizontal en 3 columnas.

El atrezzo

Para dibujar el escenario tenemos que diseñar bloques gráficos según la distancia a la que se encuentre. Para lo más alejado usaremos 2x2 caracteres como máximo, para media distancia bloques de 3x3, y para lo más cercano usaremos bloques de 3x4. Resulta evidente que podemos hacer los bloques del tamaño que deseemos, y que estas medidas sólo son orientativas.



Como podeis ver en la imagen superior, tenemos un arbol en sus tres perspectivas. No es una maravilla, pero como base nos sirve, y después podemos mejorarlos con UDG's.

La perspectiva

Que consigamos el efecto 3D deseado depende en gran medida del punto de vista que tengamos, o perspectiva.

En primer lugar vamos a definir una línea de horizonte, debajo de la cual tendremos el suelo. Según el tipo de juego debemos dibujar el suelo con un color o un dibujo. En nuestro caso vamos a pintarlo de amarillo, y el cielo de color blanco.

Un valor bueno es coger 2 caracteres de alto para el cielo y 4 de alto para el suelo, así se hizo en Crysis (parece el nombre de un libro).

Y ahora el truco maestro para ganar velocidad: dibujar los colores de fondo como papel, no como tinta, para imprimir encima con print como PAPER 8. De esta forma todo lo que dibujemos encima tendrá papel transparente, y no nos tendremos que preocupar de ir cambiando dicho valor. No es mucha la velocidad que se gana... pero ahí está.

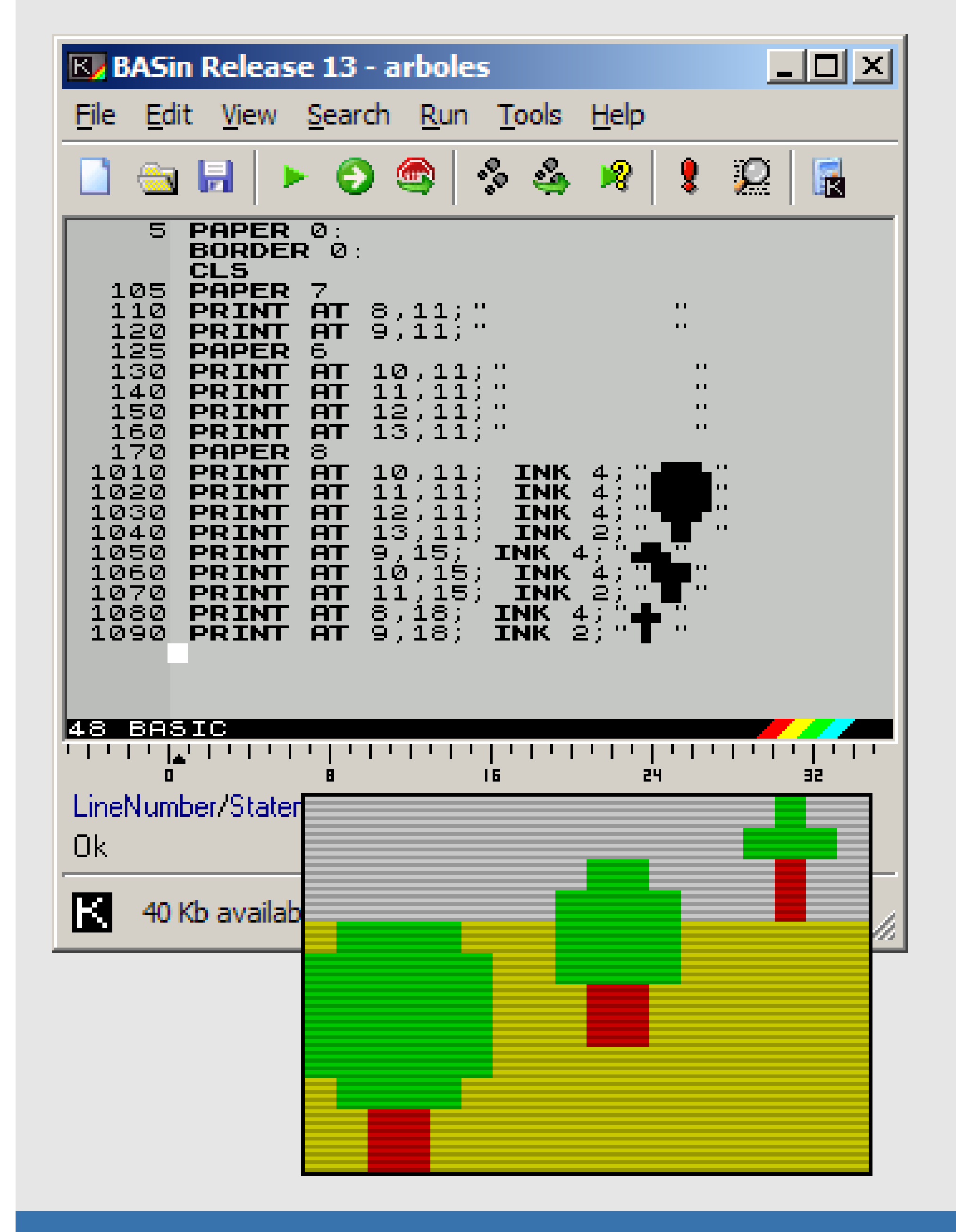
El escenario sería así:

```
10 PAPER 0: BORDER 0: CLS
100 PAPER 7
110 PRINT AT 8,11;"          "
120 PRINT AT 9,11;"          "
125 PAPER 6
130 PRINT AT 10,11;"          "
140 PRINT AT 11,11;"          "
150 PRINT AT 12,11;"          "
160 PRINT AT 13,11;"          "
170 PAPER 8
```

Bien, ya tenemos un horizonte p-e-r-f-e-c-t-o. Ahora vamos a ver como dibujar el escenario con perspectiva para crear el efecto 3D.

Los árboles más lejanos se deben dibujar justo sobre el horizonte, es decir lo más arriba posible, lo que veamos en el nivel intermedio lo dibujaremos dos caracteres por debajo, y lo que tengamos en primer plano... pues eso, en primer plano, lo más abajo posible.

Usaremos BASIN para la prueba.



Esto ya está mejor, hemos dibujado “a mano” un arbol en sus tres perspectivas, como podeis ver la posición horizontal viene determinada por dividir la visión en tres columnas de tres caracteres cada una.

En resumen, hemos dividido la pantalla de visión en 9 celdas exactamente iguales. Cada celda en vertical es lo lejos que estamos del decorado, y en horizontal el punto de vista (izquierda, centro, derecha). Debemos dibujar siempre sobre la el valor inferior de cada celda, no pasa nada si superamos el valor superior y excedemos por arriba (salvo la línea más lejana, porque excedemos el campo de visión).

Dibujando el decorado en orden

Al igual que es importante dibujar en el lugar correcto, también lo es hacerlo en el orden preciso. Y es que si dibujamos fuera de orden crearemos un bonito “collage” en pantalla.

En este caso lo importante es dibujar desde lo más alejado a lo más cercano, para que lo que veamos más cerca esté por delante. Otro truco, que sólo os servirá para un caso especial y no vamos a usar ahora, es dibujar primero lo de izquierda y derecha, dejando en último lugar la parte centrada.

Bien, ya sabemos dibujar el escenario... pero nos falta tener un mapa del mundo y cómo pasar los datos de ese mapa a una rutina que nos dibuje todo.

Diseñando el mundo

Existen dos formas de diseñar el mundo: mediante un DIM (array) o creando los datos directamente en memoria con POKE (lo cual nos facilita que podamos salvar niveles y cargarlos cuando queramos).

Vamos a explicar el método empleado en Crysis, con DIM, y luego exponemos como convertirlo al otro método.

DIM a(20,20)

Ala, ya tenemos un entorno de 20x20 posiciones. Cada posición podemos rellenarlo con un valor, que indicará el tipo de escenario que va allí. Como podeis ver es posible crear entornos muy variados, ya que con sólo usar 10 posibilidades podemos crear de todo.

Ahora vamos a definir una tabla sencilla:

0 Nada, sólo suelo
1 Árbol
2 Brizna de hierba

Rellenando el mundo

Una vez definido el mundo vamos a rellenarlo, podeis crearos vuestro propio editor o dejar las cosas a la suerte, por ejemplo con una rutina que rellene el array con 0,1, y 2 al hazar.

```
10 DIM a(20,20)
20 FOR b=1 TO 20: FOR c=1 TO 20: LET d=INT (RND*4): LET a(b,c)=d: NEXT c: NEXT b
```

Tachán, ya tenemos nuestro mapa, si quereis verlo en pantalla en 2D para ver si ha quedado bien:

```
30 FOR b=1 TO 20: FOR c=1 TO 20: PRINT AT c,b;a(c,b): NEXT c: NEXT b
```

Ojito, hemos usado un RND*4 para generar el mapa, cuando sólo usamos del 0 al 2, ¿motivo? Para no recargarlo todo y dar más sensación de movimiento, pensad que si ponemos demasiados árboles no notaremos casi el efecto de avanzar debido a que dibujamos todo por bloques fijos.

```
12100020010002010010
01002122210100011002
10111220011112201221
01020021200201020121
12201020220120211222
22012121111121201210
20022002001222120212
00201010001201210001
10002010001112101102
21101001000001110201
02212012221002020122
00220020212210221122
02022111111120200202
10222020122200110020
02221212002122022112
10011201221102112212
01202021012101221000
12101020201022100221
01000210210200021121
21010211001101211022

0 OK, 30:5
```

Si optais por realizar vuestor propio editor de mapas, que es lo más divertido que podais imaginar, es mejor que useis otro método para guardarlo: usando PEEK y POKE. Este método es igual de rápido que el anterior, pero nos permite meter todo en RAM y gestionar mejor la memoria.

En ese caso debeis reservar para nuestro ejemplo 400 bytes, bajando la RAMTOP con CLEAR, por ej.:

CLEAR 64899

El código de generación automática se cambia por:

```
10 CLEAR 64899
20 FOR b=0 TO 19: FOR c=0 TO 19: LET d=INT (RND*4): POKE (64900+(b+c*20)),d: NEXT c: NEXT b
```

La “preview” sería en este caso:

```
30 FOR b=0 TO 19: FOR c=0 TO 19: PRINT AT c,b;PEEK (64900+(b+c*20)): NEXT c: NEXT b
```

Si cronometrais vereis que no se gana en velocidad, pero para los puristas como yo se gana en orden y facilidad.

La ventaja de este método es que podemos salvar el mundo con un simple

```
SAVE “mundo.bin” CODE 64500,400
```

y recuperarlo con

```
LOAD “mundo.bin” CODE 64500,400
```

así podeis hacer múltiples niveles.

La rutina heurística 3D

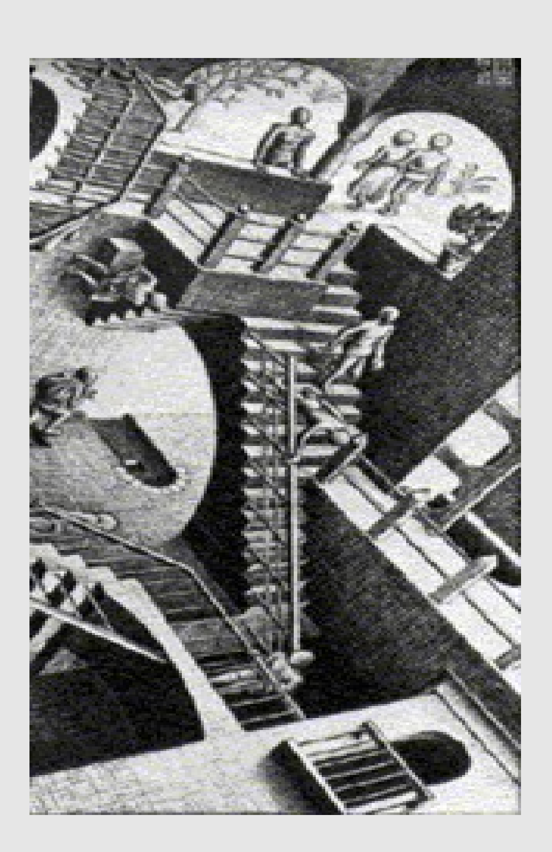
Para no convertir esto en un tratado médico, vamos a dar las pautas para moverse y dibujar el mundo en 3D de una forma simple: avanzar, retroceder, izquierda, y derecha (sin girar). Vamos a dejar los giros para un próximo artículo, al igual que los trucos que hay que realizar para dibujar también paredes y tener nuestra “mazmorra” o lo que queramos.

Los pasos para dibujar nuestro mundo son:

- Leer el teclado
- Comprobar si nos salimos del mapa
- Borrar la pantalla dibujando el horizonte
- Dibujar la fila de atrás
- Dibujar la fila intermedia
- Dibujar la fila más cercana
- Vuelta a empezar

La lectura del teclado es sencilla, vamos a usar Q A O P. Evidentemente podeis cambiar las teclas por las que más os gusten, pero estas son las más populares.

El límites de movimiento lo vamos a fijar 2 pasos en cada dirección, es decir, el rango de moviento será entre 3 y 17 pasos en vertical y horizontal, para que no intentemos leer fuera del array que define el mapa y obtengamos error.



```
100 REM inicializamos variables
110 LET x=10: LET y=10: GO TO 1000
120 REM rutina movimiento
130 LET s$=INKEY$
140 IF s$="o" AND x>3 THEN LET x=x-1:
GO TO 1000
150 IF s$="p" AND x<17 THEN LET x=x+1:
GO TO 1000
160 IF s$="q" AND y>3 THEN LET y=y-1:
GO TO 1000
170 IF s$="a" AND y<17 THEN LET y=y+1:
GO TO 1000
180 GO TO 130
```

Bien, ya tenemos nuestra rutinas de movimiento, la cual podemos añadir a la de generación del mapa. Las variables que usamos son:

X para indicar la posición horizontal en el mapa
Y para indicar la posición vertical en el mapa
s$ para guardar la tecla pulsada

La primera vez que llegamos a esta rutina, y cada vez que nos movemos, saltamos a la línea 1000, que es donde vamos a poner la rutina de dibujo de la pantalla.

Bien, ahora pasamos a la parte donde dibujamos el horizonte, que nos sirve para borrar todo.

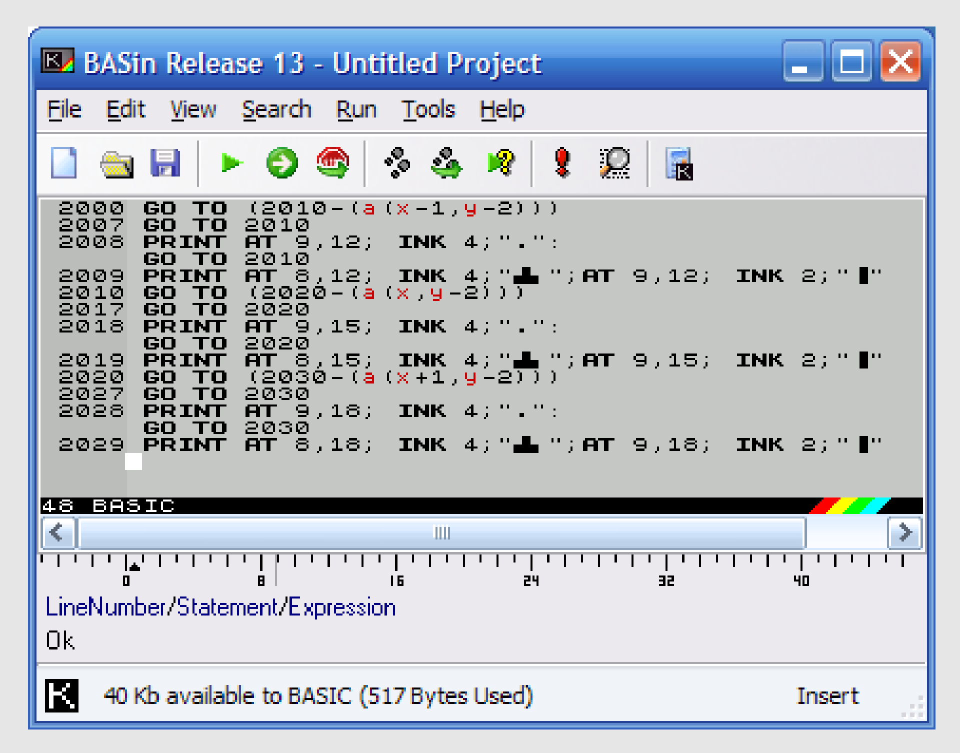
```
1000 REM rutina de vision
1010 PAPER 7
1020 PRINT AT 8,11;"         "
1030 PRINT AT 9,11;"         "
1040 PAPER 6
1050 PRINT AT 10,11;"         "
1060 PRINT AT 11,11;"         "
1070 PRINT AT 12,11;"         "
1080 PRINT AT 13,11;"         "
1090 PAPER 8
```

Como podeis ver se ha dibujado el cielo con color blanco, la tierra de amarillo, y al finalizar hemos dejado el color de papel en 8, es decir transparente. Es importante que no cambiemos el color del papel después, o bien cambiarlo pero volverlo a poner en 8.

El punto culminante, magnífico, glorioso, llega ahora. Todo el secreto de generar la imagen 3D se basa en no usar ni un solo IF para poner el decorado. ¿Cómo podemos entonces elegir el tipo de escenario según el valor del mapa?, empleando GOTO, hacemos un GOTO a la línea que dibuja el escenario y sumamos o restamos el valor del mapa.

En esta ocasión vamos a realizar lo siguiente: cada 10 líneas vamos a realizar un GOTO a la siguiente zona de pantalla a dibujar, de esta forma si restamos el valor del mapa y obtenemos 0 (nada) no dibujaremos nada y pasaremos a la siguiente zona.

Lo mejor es verlo para entenderlo, empecemos:



El código que se muestra se corresponde con la fila posterior. Como podeis ver el proceso seguido es es situar lo que se dibuja en cada posición separado por 10 líneas: 2000, 2010, y 2020. En cada una de esas líneas realizamos un salto de 10, menos el valor que se corresponde a la zona de mapa que representan, de esta forma si el valor es un 0 (nada) no se dibuja nada, si es uno dibujamos un arbol, si es 1 dibujamos un arbustillo, y si es superior a 1 realizamos un GOTO para no dibujar nada.

Esto quiere decir que no podemos añadir más de 9 tipos de escenario, si queremos más variedad tenemos que aumentar el paso entre líneas a lo que necesitemos, sólo eso.

Una ventaja clara de este sistema es que el número de tipos de escenario no merma la velocidad de la rutina, ya que sólo vamos a dibujar un objeto y pasamos de línea con un GOTO.

La secuencia completa de GOTOs para dibujar la pantalla es:

```
2000 GO TO (2010-(a(x-1,y-2)))
2010 GO TO (2020-(a(x,y-2)))
2020 GO TO (2030-(a(x+1,y-2)))
2030 GO TO (2040-(a(x-1,y-1)))
2040 GO TO (2050-(a(x,y-1)))
2050 GO TO (2060-(a(x+1,y-1)))
2060 GO TO (2070-(a(x-1,y)))
2070 GO TO (2080-(a(x,y)))
2080 GO TO (2090-(a(x+1,y)))
```

Y así de simple es, os dejo en la página web el código fuente completo, para que podais modificarlo y trastear con él con el programa BASIN. Ya no teneis excusa para realizar vuestro propio FPS o RGP con unos pocos añadidos, incluso una aventura a lo Elvira.

http://www.bytemaniacos.com/ficheros/curso_bas3d/

Creación de juegos para Interface 2

Cómo crear cartuchos para ZX Spectrum con ensamblador PASMO

Los cartuchos para Interface 2 son un invento magnífico, nos permiten disfrutar del juego de forma rápida y fiable, y además es un soporte seguro. Pero, ¿cómo realizar nuestro propio juego en este formato? El BASIC no es posible usarlo, ya que vamos a suplantar el sistema operativo del ordenador, y en lenguaje ensamblador tampoco podemos usar las rutinas básicas de la ROM.

Con motivo del nuevo concurso vamos a exponer aquí de una forma sencilla como usar el lenguaje ensamblador y el programa PASMO para generar nuestros juegos en formato cartucho.

Eso si, esto sólo sirve para gente con experiencia en el lenguaje, los demás abstenerse... o aprender.

Manos a la obra

Para crear un cartucho en lenguaje ensamblador no hace falta magia, basta con crear el código desde ORG 0, y finalizar con un NOP en 16383 para que se cree un fichero BIN de 16384 bytes (16 K exactamente) que podemos renombrar a .rom y ejecutar en cualquier emulador de ZX Spectrum para probar.

Igualmente hay que cambiar la PILA de sitio según el modelo de ZX Spectrum. Si se va a usar un 48K no hay que cambiar nada, pero si es un 16K hay que realizar un “LD SP” con la nueva dirección que queremos usar para la pila.

Yo personalmente uso:

```
ld sp, 32767
```

De esta forma, con no usar los últimos bytes de la RAM, dejamos espacio para PUSH / POP.

El esqueleto del programa quedaría:

```
; ------------------------------------------
; Prueba de ROM IF1
; ------------------------------------------

ORG 0
ld sp, 32767

; ------------------------------------------
; Aquí metemos nuestro programa
; ------------------------------------------


; ------------------------------------------
; Fin del programa
; ------------------------------------------

ORG 16383
 nop
```

Nuestro propio código ASCII

Una de las desventajas de usar el espacio de la ROM es que no podemos usar el set de caracteres del ZX Spectrum. No obstante, también podemos verlo como una ventaja, y crearnos el nuestro propio.

PASMO permite definir nuestro propio set ASCII, pero debe respetar el orden del que hay en un ZX Spectrum, y por lo tanto contener la misma información.

Peeeeero, eso no es divertido ni práctico porque desprovechamos caracteres que no usaremos y nos debemos limitar al orden impuesto y no al que nos convenga.

La diferencia es que los textos tendremos que definirlos con nuestro propio código ASCII, y no podemos usar la comodidad de poner el texto directamente entre comillas, pero a menos que vayas a realizar una aventura de texto no es problema.

```
; -----------------------------------------
; Set de caracteres para juegos en ROM
; -----------------------------------------

letra   defb 0, 124, 254, 206, 214, 230, 254, 124   ; 0 - $0
   defb 0, 28, 60, 124, 28, 28, 28, 28      ; 1 - $1
   defb 0, 252, 254, 14, 124, 224, 254, 254   ; 2 - $2
   defb 0, 252, 254, 14, 124, 14, 254, 252      ; 3 - $3
   defb 0, 198, 198, 198, 254, 126, 6, 6      ; 4 - $4
   defb 0, 254, 254, 224, 252, 14, 254, 252   ; 5 - $5
   defb 0, 124, 252, 192, 252, 198, 254, 124   ; 6 - $6
   defb 0, 254, 254, 6, 14, 28, 56, 112      ; 7 - $7
   defb 0, 124, 254, 198, 124, 198, 254, 124   ; 8 - $8
   defb 0, 124, 254, 198, 126, 6, 126, 124      ; 9 - $9
   defb 0, 124, 254, 198, 198, 254, 198, 198   ; A - $10
   defb 0, 252, 254, 198, 252, 198, 254, 252   ; B - $11
   defb 0, 126, 254, 192, 192, 192, 254, 126   ; C - $12
   defb 0, 248, 252, 206, 198, 206, 252, 248   ; D - $13
   defb 0, 126, 254, 192, 252, 192, 254, 126   ; E - $14
   defb 0, 126, 254, 192, 252, 252, 192, 192   ; F - $15
   defb 0, 124, 254, 192, 222, 198, 254, 124   ; G - $16
   defb 0, 198, 198, 198, 254, 198, 198, 198   ; H - $17
   defb 0,254, 254, 56, 56, 56, 254, 254      ; I - $18
   defb 0, 6, 6, 6, 6, 198, 254, 124      ; J - $19
   defb 0, 198, 206, 220, 248, 220, 206, 198   ; K - $20
   defb 0, 192,192, 192, 192, 192, 254, 126   ; L - $21
   defb 0, 198, 238, 254, 214, 214, 198, 198   ; M - $22
   defb 0, 198, 230, 246, 254, 222, 206, 198   ; N - $23
   defb 0, 254, 0, 230, 246, 254, 222, 206      ; Ñ - $24
   defb 0, 124, 254, 198, 198, 198, 254, 124   ; O - $25
   defb 0, 252, 254, 198, 254, 252, 192, 192   ; P - $26
   defb 0, 124, 254, 198, 214, 204, 254, 118   ; Q - $27
   defb 0, 252, 254, 198, 254, 252, 206, 198   ; R - $28
   defb 0, 126, 254, 224, 124, 14, 254, 252   ; S - $29
   defb 0, 254, 254, 56, 56, 56, 56, 56      ; T - $30
   defb 0, 198, 198, 198, 198, 198, 254, 124   ; U - $31
   defb 0, 198, 198, 198, 198, 238, 124, 56   ; V - $32
   defb 0, 198, 198, 214, 214, 254, 254, 108   ; W - $33
   defb 0, 198, 238, 124, 56, 124, 238, 198   ; X - $34
   defb 0, 198, 198, 254, 126, 6, 126, 124      ; Y - $35
   defb 0, 254, 254, 28, 56, 112, 254, 254      ; Z - $36
   defb 0, 0, 0, 254, 254, 0, 0, 0         ; - - $37
   defb 0, 24, 24, 126, 126, 24, 24, 0      ; + - $38
   defb 0, 254, 254, 0, 0, 0, 254, 254      ; = - $39
   defb 0, 6, 14, 28, 56, 112, 224, 192      ; / - $40
   defb 0, 108, 108, 108, 0, 0, 0, 0      ; " - $41
   defb 0, 0, 0, 0, 0, 0, 96, 96         ; . - $42
   defb 0, 0, 0, 0, 0, 0, 64, 32         ; , - $43
   defb 0, 0, 96, 96, 0, 0, 96, 96         ; : - $44
   defb 0, 0, 192, 192, 0, 0, 128, 64      ; ; - $45
   defb 0, 96, 48, 24, 12, 24, 48, 96      ; > - $46
   defb 0, 12, 24, 48, 96, 48, 24, 12      ; < - $47
   defb 0, 24, 56, 48, 48, 48, 56, 24      ; ( - $48
   defb 0, 48, 56, 24, 24, 24, 56, 48      ; ) - $49
   defb 0, 24, 24, 24, 24, 0, 24, 24      ; ! - $50
   defb 0, 56, 124, 108, 12, 24, 0, 24      ; ? - $51
   defb 0, 198, 206, 28, 56, 112, 230, 198      ; % - $52
```

Tachán! Ya tenemos nuestro propio set de caracteres con lo básico para un juego normal. Lo bueno de este sistema es que ahorramos memoria y podemos usar caracteres nuevos (como la Ñ).

Para definir una frase debemos usar el valor ASCII directamente, ej:

```
frase  defb 11, 35, 30, 14, 22, 10, 23,
18, 10, 12, 25, 29, 42, 12, 25, 22, 255
```

Nuestras propias rutinas

Es importante, a la hora de implementar nuestras rutinas, recordar que no tenemos una ROM presente. Eso quiere decir que no podemos usar ni rutinas de la ROM ni las variables del sistema (porque no existen).

Por ello vamos a ir dando algunas rutinas básicas para salir del paso y empezar a programar.

```
; ------------------------------------------
; CLS SIN USAR LA ROM
; ENTRADAS: a, es el color de tinta/papel
; SALIDAS: pone el mapa de pantalla a 0
; ------------------------------------------

sucls   ld hl, 22528
   ld de, 22529
   ld bc, 767
   ld (hl),a
   ldir
   ld hl, 16384
   ld de, 16385
   ld bc, 6143
   ld (hl),l
   ldir
   ret
```

La rutina anterior nos realizará un borrado de toda la pantalla, con los colores de tinta y papel que metamos en el acumulador según la tabla:

```
 - TINTA -
NEGRO      0   00000000
AZUL       1   00000001
ROJO       2   00000010
MAGENTA    3   00000011
VERDE      4   00000100
AZULADO    5   00000101
AMARILLO   6   00000110
BLANCO     7   00000111

- PAPEL -
NEGRO      0   00000000
AZUL       8   00001000
ROJO      16   00010000
MAGENTA   24   00011000
VERDE     32   00100000
AZULADO   40   00101000
AMARILLO  48   00110000
BLANCO    56   00111000
```

Basta con sumar el valor de tinta y papel para obtener el valor que hay que introducir en el acumulador antes de llamar a la rutina.

Para cambiar el borde de la pantalla basta un OUT:

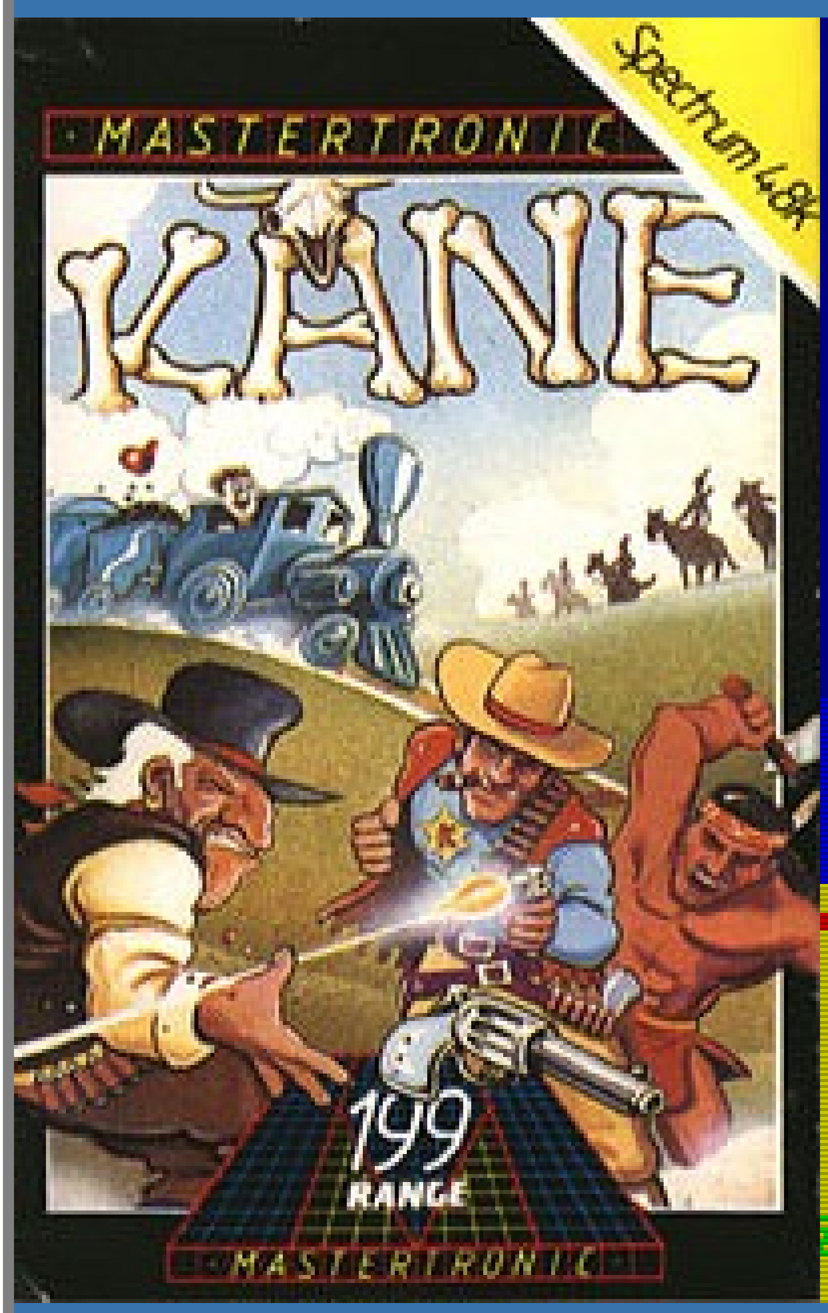
```
; ------------------------------------------
; CAMBIAR EL BORDE DE PANTALLA
; ENTRADAS: a, es el color del borde
; SALIDAS: ninguna
; ------------------------------------------

borde   out (254), a
   ret
```

En el próximo número veremos como realizar sprites, al igual que rutinas para imprimir nuestro propio código ASCII de una forma sencilla.

Si os perdeis, o teneis alguna duda, podeis preguntar en los foros de programación de Speccy.org. De hecho en ellos encontrareis tres hilos con el nombre “Creando cartuchos IF2 con PASMO”, en los que podeis ver mejor los listados aquí expuesto y copiarlos con comodidad a vuestro editor favorito.

A programar!



Variedad

El sello Mastertronic supuso mucho para los jóvenes de entonces, sabias que un juego con ese sello hiba a ser divertido, gozaría de una portada currada, y tendría un precio muy sugerente.

Kane era todo eso, un juego divertido con varios niveles distintos entre si que formaban un conjunto realmente bueno. Destaca el movimiento del caballo del protagonista, realista y suave, además del excelente control.

En total hay tres tipos de juego en uno: disparar pájaros, correr a caballo (bien en el desierto o detrás de un tren), y defenderte en el pueblo a disparo limpio. Lo mejor del Western en un ZX Spectrum 48K.

